eurostat: Eurostat Open Data R Tools DRAFT VERSION IN PROGRESS

Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek

Abstract Governmental institutions are increasingly opening up their data resources for the public as open data. This is providing novel opportunities for research and citizen science, but efficient tools to access and analyze these data sets are needed to realize the full potential of the new information resources. We introduce the **eurostat** R package that provides a suite of tools to access open data from Eurostat, including functions to search, download, and manipulate Eurostat data in an automated and reproducible manner. The online documentation provides detailed examples on how to access, summarize and visualize these spatio-temporal data sets. The package expands previous related work and has been extensively tested by the user community. This contributes to the growing ecosystem of R packages that provide algorithmic tools for reproducible computational research in social science and humanities.

Introduction

Eurostat, the statistical office of the European Union, provides a rich collection of demographic and economic data through its open data service, which includes thousands of data sets on European demography, economics, health, infrastructure, traffic and other topics. The statistics are often available with great geographical resolution and including time series spanning over several years or decades.

The availability of tools to access and analyse such data collections can greatly benefit reproducible research (Gandrud, 2013; Boettiger et al., 2015). When the data resources and analysis algorithms are openly available, the complete analytical workflow spanning from raw data to the final publication can be made fully transparent. Standardization of common data analysis tasks via dedicated software packages can help to automate the analysis workflow, greatly facilitating reproducibility and code sharing, and making the data analysis more efficient. The algorithms need to be customized to specific data sources, however, to accommodate variations in raw data formats, access details, and typical use cases so that the end user can avoid repetitive standard programming tasks and spend more time on the actual research. A number of packages to access open data from governmental and other institutions have been consequently designed to meet these demands and to access open data from the Food and Agricultural Organization (FAO) of the United Nations (**FAOSTAT**; Kao et al. (2015)), World Bank (**WDI**; Arel-Bundock (2013)), national statistics authorities (**pxweb**; Magnusson et al. (2014)), Open Street Map (**osmar**; Eugster and Schlesinger (2012)) and many other sources.

A dedicated R package for eurostat open data has been missing, however. We introduce the **eurostat** R package to fill this gap and to facilitate automated access to open data from Eurostat[1]. This brings together our earlier efforts with the **statfi** (Lahti et al., 2013) and **smarterpoland** (Biecek, 2015) packages. We have combined the relevant parts of these two packages and implemented an expanded set of tools with a specific focus on the Eurostat data collection. Since its first CRAN release in 2014, the package has been actively developed by several contributors and based on community feedback in Github. We are now reporting the first mature version that has been improved and tested by multiple users. The package and its predecessors have been applied in several case studies by us and others[2].

The **datamart** (Weinert, 2014), the **quandl** (McTaggart et al., 2015) and the **pdfetch** (Reinhart, 2015) R packages provide further functions that can be used to access certain versions of Eurostat data. In contrast to these generic database packages, our eurostat package provides functionality that is particularly tailored for the Eurostat open data service. The **eurostat** package greatly benefits from further tools in the **dplyr** (Wickham and Francois, 2015), **knitr** (Xie, 2015), **ggplot2** (Wickham, 2009), **mapproj** (for R by Ray Brownrigg et al., 2015), and **stringi** (Gagolewski and Tartanus, 2015) R packages. The **eurostat** package is part of the rOpenGov collection (Leo Lahti and Kainu, 2013) that provides reproducible research tools for computational social science and digital humanities.

In summary, the **eurostat** package provides custom tools to search, retrieve, modify and visualize data from the Eurostat open data service. The package supports key features such as data cache, date formatting, and tidy data principles (Wickham, 2014) using the the **tidyr** R package (Wickham, 2015c). Here, we provide an overview of the core functionality in the current CRAN release version (1.2.1). For further documentation and the reproducible source code for this article, see the package Github site [3].

[1] http://ec.europa.eu/eurostat/data/database

[2] See e.g. http://blog.revolutionanalytics.com/2015/04/financial-times-tracks-unemployment-with-r.html

[3] https://github.com/rOpenGov/eurostat

Search and download commands

To install and load the CRAN release version, just type in R:

```
> install.packages("eurostat")
> library("eurostat")
```

The complete table of contents of the database can be browsed on-line[4], or downloaded in R with the command `toc <-get_eurostat_toc()`. The function `search_eurostat()` is used to make a more focused search over the table of contents. To retrieve data for 'road accidents', for instance, use:

```
> query <- search_eurostat("road accidents", type = "table")
```

The `type` argument limits the search on a selected data set type in the above example. The options for this argument include *'table'*, *'dataset'* or *'folder'*, referring to different levels of hierarchy in the data organization: a `table` resides in `dataset`, which is in turn stored in a `folder`.

Values in the `code` column of the `search_eurostat()` function output provide data sets identifiers that can be used in subsequent download commands. Alternatively, these identifier codes can be browsed at the Eurostat open data service; check the codes in the Data Navigation Tree listed after each dataset in parentheses. Let us look at the data set identifier and title for the first entry of the query data:

```
> query$code[[1]]
[1] "tsdtr420"

> query$title[[1]]
[1] "People killed in road accidents"
```

Let us next retrieve the data set with this identifier as follows:

```
> dat <- get_eurostat(id = "tsdtr420", time_format = "num")
```

As the original data is annual in this example, we have selected a numeric time format. This is more convient for annual time series than the default date format. The data sets are provided as standard data frames to support standard tools for data subsetting and reshaping. The above function call returns a table on transport statistics. The first lines of the output are shown in Table 1. Visualization with the following commands reveals a decreasing trend of road accidents in many countries over time (Figure 1):

```
t1 <- get_eurostat("tsdtr420") %>%
  dplyr::filter(geo %in% c("UK", "SK", "FR", "PL", "ES", "PT", "LV"))
t1$Country <- t1$geo
ggplot(t1, aes(x = time, y = values, color=Country, group=Country, shape=Country)) +
  geom_point(size=4) +
  geom_line() + theme_bw() + ggtitle("Road accidents") +
  xlab("Year") + ylab("Victims (n)") + theme(legend.position="top")
```

Utilities

	sex	geo	time	values
1	T	AT	1999.00	1079.00
2	T	BE	1999.00	1397.00
3	T	BG	1999.00	
4	T	CH	1999.00	
5	T	CY	1999.00	
6	T	CZ	1999.00	1455.00

Table 1: First lines of output from the `get_eurostat()` function with the road accident data set identifier 'tsdtr420'.

[4] http://ec.europa.eu/eurostat/data/database

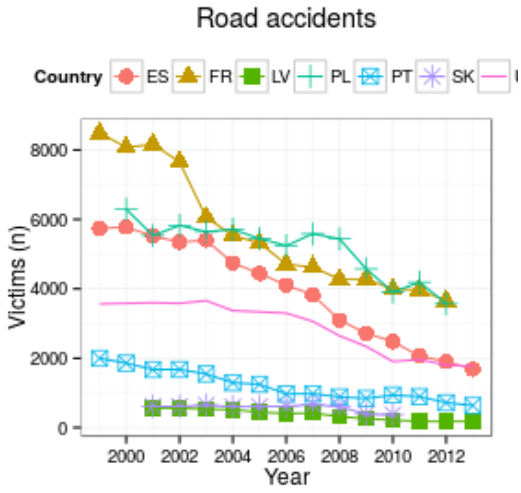

Figure 1: Timeline indicating the number of people killed in road accidents in various countries based on data retrieved with the **eurostat** package.

Many entries in Table 1 are not readily interpretable, but a simple call `label_eurostat(dat)` converts the original identifier codes into human-readable versions (shown in Table 2) based on translations in the Eurostat database.

The downloaded data sets are stored in cache by default to avoid repeated downloads of identical data sets. This can speed up the analysis. Storing an exact copy of the retrieved raw data on the hard disk supports also the reproducibility when the source database is constantly updated.

The Eurostat database includes a variety of demographic and health indicators. We see, for instance, that overweight varies remarkably across different age groups quantified by the body-mass index (BMI) (Figure 2 A). Sometimes the data from the eurostat database requires more complex pre-processing. Let's consider a question about distribution of sources of renewable energy in different European countries. In order to summarise such sources one needs to first aggregate all possible sources into a smaller number of interesting groups. Then with the use of packages like **dplyr** or **tidyr** one can process data, chop country names, filter countries depending on production levels, normalize the within country production. After a series of such transformations (see Appendix for the source code) we can finally plot the data to discover that countries vary a lot in terms of sources of renewable energy (Figure 2 B). Three-dimensional data sets such as this can be conveniently visualized as triangular maps by using the **plotrix** (Lemon, 2006) package.

Geospatial information

Map visualizations

The indicators in the Eurostat open data service are typically available as annual time series grouped by country, and sometimes at more refined temporal or geographic levels. Eurostat provides complementary geospatial data on the corresponding administrative statistical units to support visualizations

	sex	geo	time	values
1	Total	Austria	1999.00	1079.00
2	Total	Belgium	1999.00	1397.00
3	Total	Bulgaria	1999.00	
4	Total	Switzerland	1999.00	
5	Total	Cyprus	1999.00	
6	Total	Czech Republic	1999.00	1455.00

Table 2: The output from `get_eurostat()` (Table 1), now converted into human-readable labels by `label_eurostat()`.

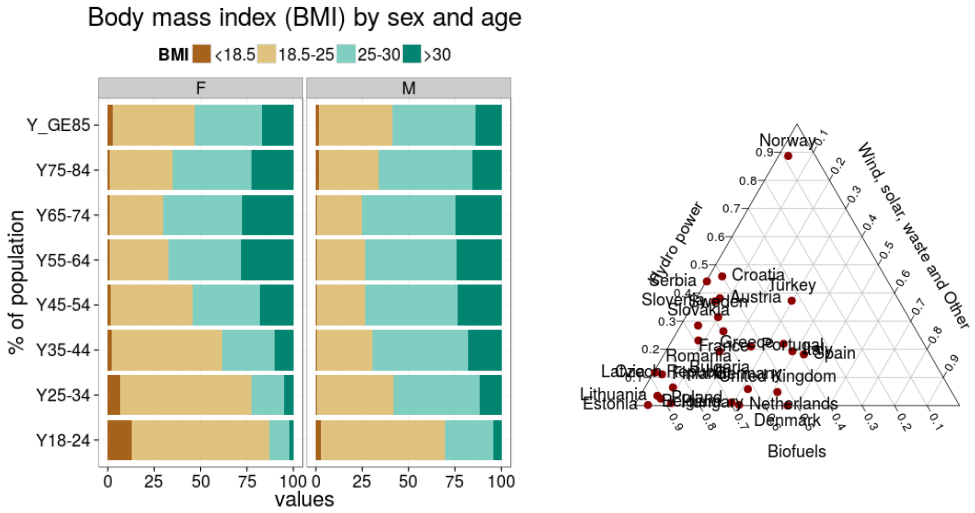

Figure 2: **A** The body-mass index in different age groups in Poland based on Eurostat table 'hlth_ehis_de1'. **B** Distribution of sources of renewable energy production based on Eurostat table `ten00081`, year 2013. See the Appendix for the source code for both figures.

at the appropriate geographic resolution. The geospatial data sets are available as standard shapefiles[5]. As an example, let us look at disposable income of private households (data set identifier tgs00026[6]). This information is provided at the geographic level of NUTS2 regions. This is the intermediate level of territorial units in the Eurostat regional classifications, and roughly corresponds to provinces or states in each country[7] (Figure 3). The example demonstrates how the Eurostat data sets and geospatial data, retrieved with the eurostat package, can be combined with additional visualization tools and other utilities including **grid** (R Core Team, 2015), **maptools** (Bivand and Lewin-Koh, 2015), **rgdal** (Bivand et al., 2015), **rgeos** (Bivand and Rundel, 2015), **scales** (Wickham, 2015a), and **stringr** (Wickham, 2015b).

Default country groupings

To facilitate further analysis and visualization of standard European country groups, we have included ready-made country code lists. The list of EFTA countries is retrieved, for instance, with:

```
data(efta_countries)
```

This provides the EFTA country listing in Table 3. Similar lists are available for Euro area (ea_countries), EU (eu_countries) and the EU candidate countries (candidate_countries). These auxiliary data sets facilitate the selection of specific country groups in the analysis. The full name and a two-letter identifier are provided for each country as provided by the Eurostat database. The country codes follow the ISO 3166-1 alpha-2 standard, except that GB and GR are replaced by UK (United Kingdom) and EL (Greece) in the Eurostat database, respectively. Linking these country codes with external data sets can be facilitated by conversions between different country coding standards with the **countrycode** package (Arel-Bundock, 2014).

[5] http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units
[6] http://ec.europa.eu/eurostat/en/web/products-datasets/-/TGS00026
[7] http://ec.europa.eu/eurostat/web/nuts/overview

	code	name
1	IS	Iceland
2	LI	Liechtenstein
3	NO	Norway
4	CH	Switzerland

Table 3: The EFTA country listing from the eurostat R package.

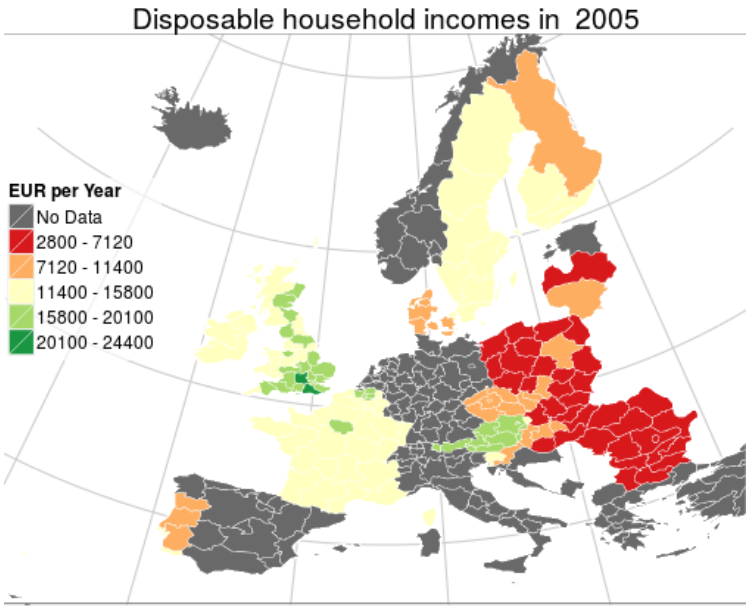

Figure 3: Disposable income of private households across NUTS2-level national regions in European countries visualized based on geospatial data available from Eurostat.

Summary

The **eurostat** R package provides convenient tools to access open data from Eurostat. Combining programmatic access to the data sets with further analysis and visualization tools allows a seamless and reproducible automation of the complete data analytical workflow from accessing the raw data to statistical analysis and final publication. The source code and installation instructions for the latest development version of the eurostat package are available at the github site, as well as the full source code of the figures and tables of this manuscript[8], where the Rmarkdown document provides reproducible documentation with full algorithmic details on the analyses, and can be updated when new versions of the Eurostat data become available.

The eurostat package provides one example of automated data retrieval from institutional data repositories, featuring options such as search, subsetting and cache. Possible future extensions and improvements include implementation of specific data representation formats to harmonize the data representation across similar data sources and to facilitate subsequent tool development. In particular, we should take further advantage of the existing spatiotemporal data structures available in R, such as those provided by the **spacetime** package (Pebesma, 2012), and construct wrapper functions to speed up routine operations such as visualizing the temporal and geospatial data sets from Eurostat. The package source code can be freely used, modified and distributed under the BSD-2-clause (modified FreeBSD) license. We welcome issues, bug reports and other feedback.

Acknowledgements

We are grateful to Eurostat for maintaining the open data service and the rOpenGov[9] for supporting R package development. This work has been partially funded by Academy of Finland (decision 293316). We also wish to thank Juuso Parkkinen and Joona Lehtomäki for feedback.

[8] https://github.com/rOpenGov/eurostat
[9] https://github.ropengov.io

Bibliography

V. Arel-Bundock. *WDI: World Development Indicators (World Bank)*, 2013. URL http://CRAN.R-project.org/package=WDI. R package version 2.4. [p1]

V. Arel-Bundock. *countrycode: Convert Country Names and Country Codes*, 2014. URL http://CRAN.R-project.org/package=countrycode. R package version 0.18. [p4]

P. Biecek. *SmarterPoland: Tools for Accessing Various Datasets Developed by the Foundation SmarterPoland.pl*, 2015. URL http://CRAN.R-project.org/package=SmarterPoland. R package version 1.5. [p1]

R. Bivand and N. Lewin-Koh. *maptools: Tools for Reading and Handling Spatial Objects*, 2015. URL http://CRAN.R-project.org/package=maptools. R package version 0.8-37. [p4]

R. Bivand and C. Rundel. *rgeos: Interface to Geometry Engine - Open Source (GEOS)*, 2015. URL http://CRAN.R-project.org/package=rgeos. R package version 0.3-14. [p4]

R. Bivand, T. Keitt, and B. Rowlingson. *rgdal: Bindings for the Geospatial Data Abstraction Library*, 2015. URL http://CRAN.R-project.org/package=rgdal. R package version 1.0-7. [p4]

C. Boettiger, S. Chamberlain, E. Hart, and K. Ram. Building software, building community: Lessons from the ropensci project. *Journal of Open Research Software*, 3(1), November 2015. [p1]

M. J. A. Eugster and T. Schlesinger. Openstreetmap and r. *R Journal*, 5(1):53–63, June 2012. [p1]

D. M. P. for R by Ray Brownrigg, T. P. Minka, and transition to Plan 9 codebase by Roger Bivand. *mapproj: Map Projections*, 2015. URL http://CRAN.R-project.org/package=mapproj. R package version 1.2-4. [p1]

M. Gagolewski and B. Tartanus. *R package stringi: Character string processing facilities*, 2015. URL http://stringi.rexamine.com/. [p1]

C. Gandrud. *Reproducible Research with R and R Studio*. Chapman & Hall/CRC, July 2013. [p1]

M. C. J. Kao, M. Gesmann, and F. Gheri. *FAOSTAT: Download Data from the FAOSTAT Database of the Food and Agricultural Organization (FAO) of the United Nations*, 2015. URL http://CRAN.R-project.org/package=FAOSTAT. R package version 2.0. [p1]

L. Lahti, J. Parkkinen, and J. Lehtomäki. statfi r package, 2013. [p1]

J. Lemon. Plotrix: a package in the red light district of r. *R-News*, 6(4):8–12, 2006. [p3]

J. L. Leo Lahti, Juuso Parkkinen and M. Kainu. ropengov: open source ecosystem for computational social sciences and digital humanities. Presentation at ICML/MLOSS workshop (Int'l Conf. on Machine Learning - Open Source Software workshop)., December 2013. URL http://ropengov.github.io. [p1]

M. Magnusson, L. Lahti, and L. Hansson. pxweb: R tools for px-web api, 2014. URL http://CRAN.R-project.org/package=pxweb. R package version 0.5.57. [p1]

R. McTaggart, G. Daroczi, and C. Leung. *Quandl: API Wrapper for Quandl.com*, 2015. URL http://CRAN.R-project.org/package=Quandl. R package version 2.7.0. [p1]

E. Pebesma. spacetime: Spatio-temporal data in r. *Journal of Statistical Software*, 51(7):1–30, 2012. URL http://www.jstatsoft.org/v51/i07/. [p5]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL https://www.R-project.org/. [p4]

A. Reinhart. *pdfetch: Fetch Economic and Financial Time Series Data from Public Sources*, 2015. URL http://CRAN.R-project.org/package=pdfetch. R package version 0.1.7. [p1]

K. Weinert. *datamart: Unified access to your data sources*, 2014. URL http://CRAN.R-project.org/package=datamart. R package version 0.5.2. [p1]

H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009. ISBN 978-0-387-98140-6. URL http://had.co.nz/ggplot2/book. [p1]

H. Wickham. Tidy data. *Journal of Statistical Software*, 59(10), 2014. [p1]

H. Wickham. *scales: Scale Functions for Visualization*, 2015a. URL http://CRAN.R-project.org/package=scales. R package version 0.3.0. [p4]

H. Wickham. *stringr: Simple, Consistent Wrappers for Common String Operations*, 2015b. URL http://CRAN.R-project.org/package=stringr. R package version 1.0.0. [p4]

H. Wickham. *tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions*, 2015c. URL http://CRAN.R-project.org/package=tidyr. R package version 0.3.1. [p1]

H. Wickham and R. Francois. *dplyr: A Grammar of Data Manipulation*, 2015. URL http://CRAN.R-project.org/package=dplyr. R package version 0.4.3. [p1]

Y. Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2015. URL http://yihui.name/knitr/. R package version 1.11. [p1]

Leo Lahti
Department of Mathematics and Statistics
PO Box 20014 University of Turku
Finland
leo.lahti@iki.fi

Janne Huovari
Pellervo Economic Research PTT
Eerikinkatu 28 A 00180 Helsinki
Finland
janne.huovari@ptt.fi

Markus Kainu
Affiliation
Address
Country
author3@work

Przemyslaw Biecek
Faculty of Mathematics, Informatics, and Mechanics
University of Warsaw
Banacha 2, 02-097 Warsaw
Poland
P.Biecek@mimuw.edu.pl

Appendix

Source code for the obesity example (Figure 2 A):

```
library(dplyr)
tmp1 <- get_eurostat("hlth_ehis_de1", time_format = "raw")
tmp1 %>%
  dplyr::filter( isced97 == "TOTAL" ,
          sex != "T",
          age != "TOTAL", geo == "PL") %>%
  mutate(BMI = factor(bmi,
                      levels=c("LT18P5","18P5-25","25-30","GE30"),
                      labels=c("<18.5", "18.5-25", "25-30",">30"))) %>%
  arrange(BMI) %>%
  ggplot(aes(y=values, x=age, fill=BMI)) +
  geom_bar(stat="identity") +
  facet_wrap(~sex) + coord_flip() +
  theme(legend.position="top") + ggtitle("Body mass index (BMI) by sex and age")+xlab("\% of population")+sca
```

Source code for the renewable energy example (Figure 2 B):

```
# All sources of renewable energy are to be grouped into three sets
> dict <- c("Solid biofuels (excluding charcoal)" = "Biofuels",
+          "Biogasoline" = "Biofuels",
+          "Other liquid biofuels" = "Biofuels",
+          "Biodiesels" = "Biofuels",
+          "Biogas" = "Biofuels",
+          "Hydro power" = "Hydro power",
+          "Tide, Wave and Ocean" = "Hydro power",
+          "Solar thermal" = "Wind, solar, waste and Other",
+          "Geothermal Energy" = "Wind, solar, waste and Other",
+          "Solar photovoltaic" = "Wind, solar, waste and Other",
+          "Municipal waste (renewable)" = "Wind, solar, waste and Other",
+          "Wind power" = "Wind, solar, waste and Other",
+          "Bio jet kerosene" = "Wind, solar, waste and Other")
# Some cleaning of the data is required
> energy3 <- get_eurostat("ten00081") %>%
+  label_eurostat(dat) %>%
+  filter(time == "2013-01-01",
+         product != "Renewable energies") %>%
+  mutate(nproduct = dict[as.character(product)], # just three categories
+         geo = gsub(geo, pattern=" \\(.*", replacement="")) %>%
+  select(nproduct, geo, values) %>%
+  group_by(nproduct, geo) %>%
+  summarise(svalue = sum(values)) %>%
+  group_by(geo) %>%
+  mutate(tvalue = sum(svalue),
+         svalue = svalue/sum(svalue)) %>%
+  filter(tvalue > 1000,
+         !grepl(geo, pattern="^Euro")) %>% # only large countrie
+  spread(nproduct, svalue)
# Triangle plot
> library(plotrix)
> par(cex=0.75)
> plotrix::triax.plot(as.matrix(energy3[, c(3,5,4)]),
+                     show.grid = TRUE,
+                     label.points = TRUE, point.labels = energy3$geo,
+                     pch = 19)
```